ECE 375 LAB 1

Introduction to AVR Development Tools

Lab Time: Friday 4 - 6

Jordan Brown

Faaiq Waqar

ADDITIONAL QUESTIONS.

1) What specific font is used for source code, and at what size?

The font used for source code is font Courier New in size 8 font

2) What is the naming convention for source code (asm)? What is the naming convention for source code files if you are working with your partner?

The naming convention for a single user is Firstname_Lastname_Lab#_sourcecode.asm. If you happen to be working with a partner, the format for your filename will be FirstnamePartner1_LastnamePartner 1_and_FirstnamePartner2_LastnamePartner2_Lab#_sourcecode.asm.

3) What is the difference between the .def and .equ directives?

By observation, .def is used for register and counter assignment declarations. .equ is used for bit assignments for associated variables. If we use the AVR Assembler directives, we see .def used for defining a symbolic name on a register, where .equ is used to set a symbol equal to an expression.

4) From that explanation, determine the 8-bit binary value that each of the following expressions evaluates to

(a) (1 << 3)

Translation to Binary: b00001000, this translates to 8 in decimal

(b) (2 << 2)

Translation to Binary: 00000010 shifted 2 bits to the left becomes 00001000. This now represents 8 in decimal

(c) (8 >> 1)

Translation to Binary: 00001000 shifted 1 bit to the left becomes 00000100, which represents 4 in decimal

(d) (1 << 0)

Translation to Binary: 00000000 does not change when shifted because all bits are 0's, it is still 00000000.

(e) (6 >> 1|1 << 6)

1 shifted right by 6: 01000000

1 shifted left by 6: 00000100

final result: 01000100, this represents 68 in decimal

5) describe the instructions listed below. ADIW, BCLR, BRCC, BRGE, COM, EOR, LSL, LSR, NEG, OR, ORI, ROL, ROR, SBC, SBIW, and SUB

ADIW: Add Immediate to Word

Brief Description: Adds an immediate value ranging from 0-63 to a register pair and places that are a resultant of the register pair. ADIW operates on the upper 4 register pairs, and is suited for use with pointer registers.

BCLR: Bir Clear in SREG

Brief Description: Used to clear a single flag in the status register

BRBC: Branch if Bit in SREG is Cleared

Brief Description: This is a conditional relative branch that tests a single bit in the status register and branches relative to the program counter if the bit is cleared. The offset parameter is represented in two's complement form.

BRGE: Branch if Greater or Equal (Signed)

Brief Description: This is a conditional relative branch. It tests the signed flag and branches relative to the program counter if the signed flag is cleared. If this instruction is used right after CP, CPI, SUB or SUBI, the branch will occur if and only if the signed binary number in register Rd is greater than or equal to the signed binary number i register Rr.

COM: One's Complement

Brief Description: Performs a One's complement of register Rd

EOR: Exclusive Or

Brief Description: Performs a logical exclusive or between the contents of register Rd and register Rr and places the output result in register Rd

LSL: Logical Shift Left

Brief Description: Shift all the bits in register Rd one place to the left, where bit 0 is cleared, bit 7 is loaded into the C flag of the status register, and the operation multiplies signed and unsigned vals by two

LSR: Logical Shift Right

Brief Description: Shift all the bits in register Rd one place to the left, where bit 7 is cleared, bit 0 is loaded into the C flag of the status register, and the operation divides unsigned vals by two

NEG: Two's Complement

Brief Description: Replaces the contents of register Rd with its two's complement

OR: Logical OR

Brief Description: Performed a logical Or operation between the contents of register Rd and Rr, and places the result in Rd

ORI: Logical OR with Immediate

Brief Description: Performs a logical Or between contents of register Rd and a constant, and places the result in Rd

ROL: Rotate Left through Carry

Brief Description: Shift all the bits in register Rd one place to the left. C flag is shifted into bit 0 of register Rd, and bit 7 is shifted into the C flag. This effectively multiplies multi-byte signed and unsigned values by two.

ROR: Rotate Right through Carry

Brief Description: Shift all the bits in register Rd one place to the left. C flag is shifted into bit 7 of register Rd, and bit 0 is shifted into the C flag. This effectively divides multi-byte signed values by two.

SBC: Subtract with Carry

Brief Description: Subtracts two registers and subtracts with the C flag, and places resultant into register Rd

SBIW: Subtract Immediate from Word

Brief Description: Subtracts an immediate value from 0-63 from a register pair and places the result in the register pair

SUB: Subtract Without Carry

Brief Description: Subtracts two registers and places the result in destination register Rd